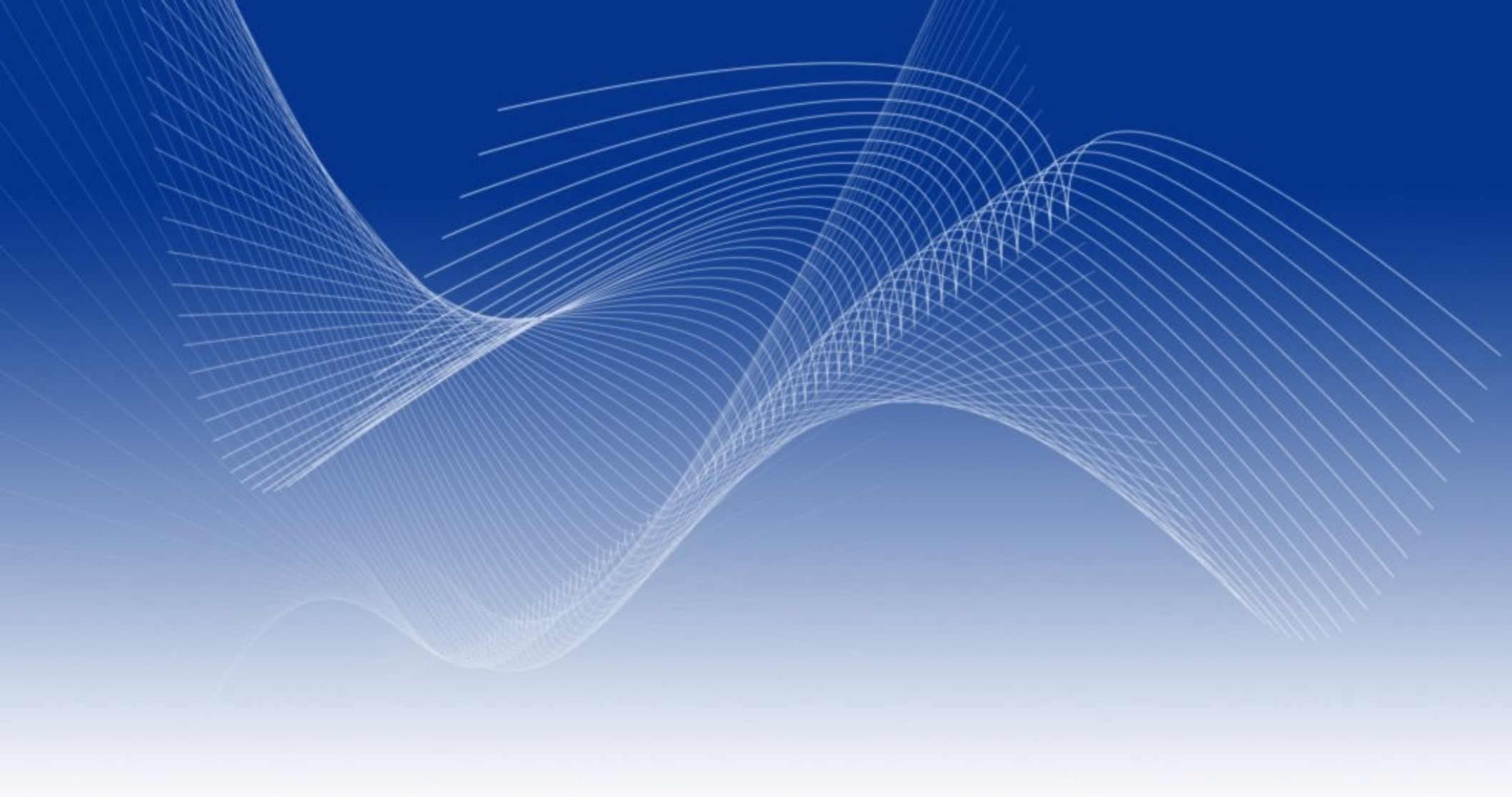


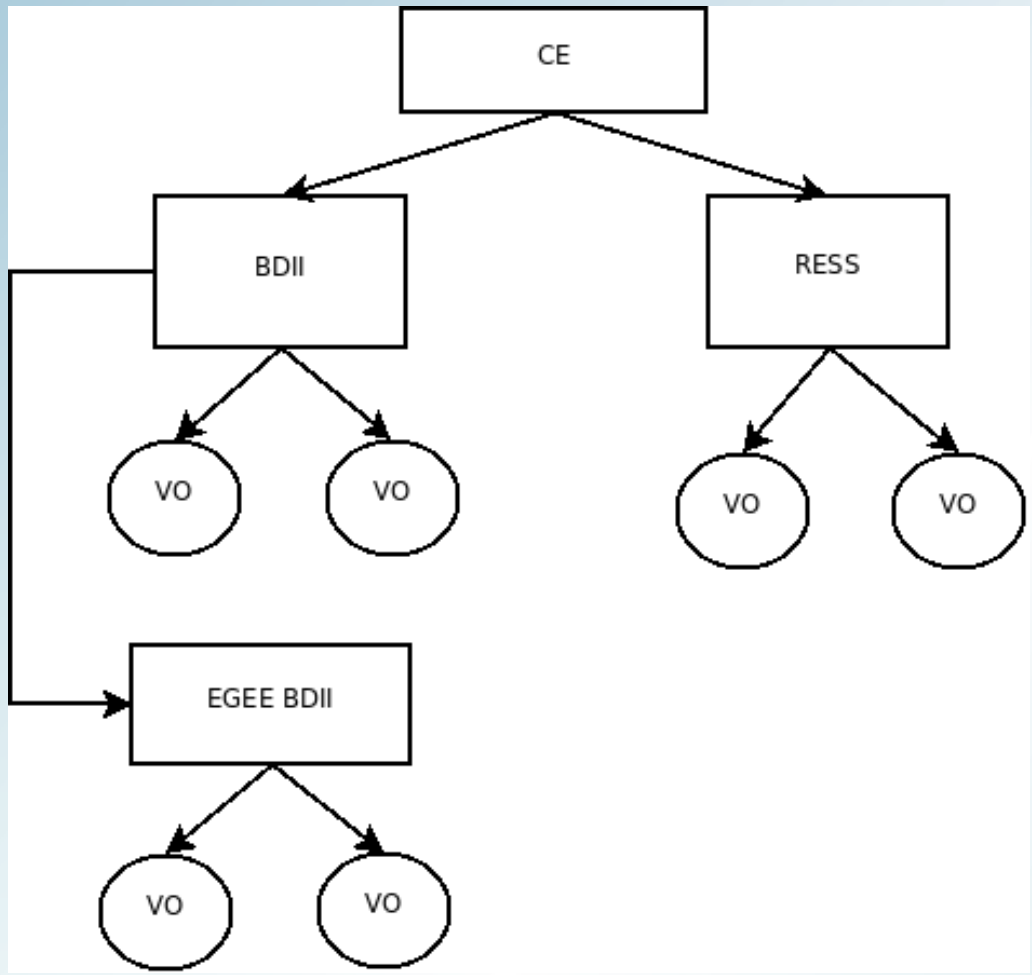
End to End Information Infrastructure & Services in Open Science Grid (OSG)

Anthony Tiradani
Fermilab

Definitions and Scope

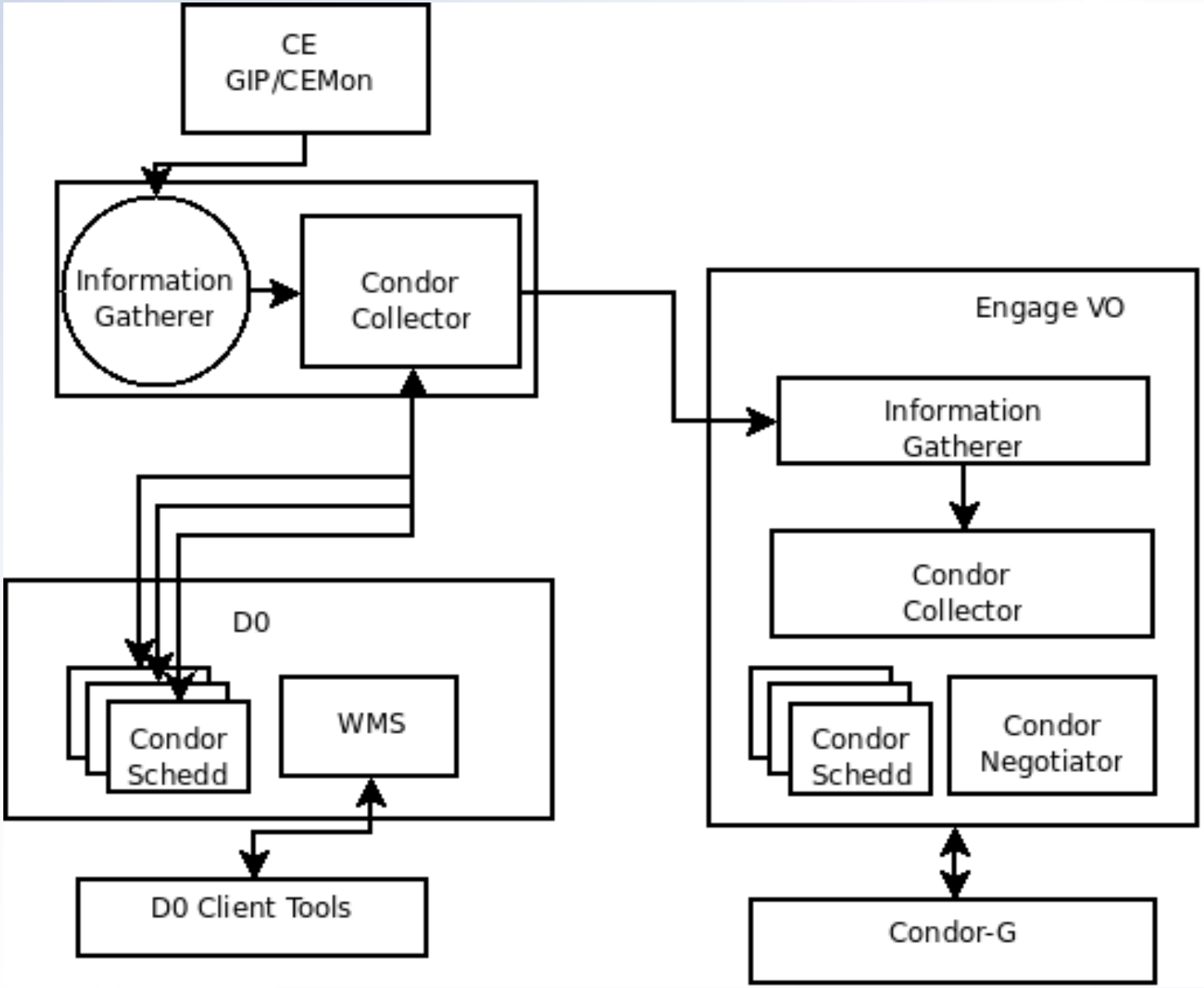
- Information Types:
  - Monitoring: Information that gives a view on resource health and current availability. Example: RSV
  - Accounting: Information that gives a view of what resources are being/has been used, for how long, and by whom. Example: Gratia
  - Discovery: Information that provides a view of what services are offered and what those services look like. Example: RESS/BDII
- This discussion will deal with Discovery information. How is it generated and published? Who consumes it?
- Note: The current implementation of the Discovery Information contains data that can cross definitions. There are groups that use it for other purposes (Installed Capacity).

Current Information Flow



- CE publishes (pushes) information to a BDII Collector
- CE also publishes (pushes) information to RESS Collector
- VO's access RESS or the BDII for service/resource discovery

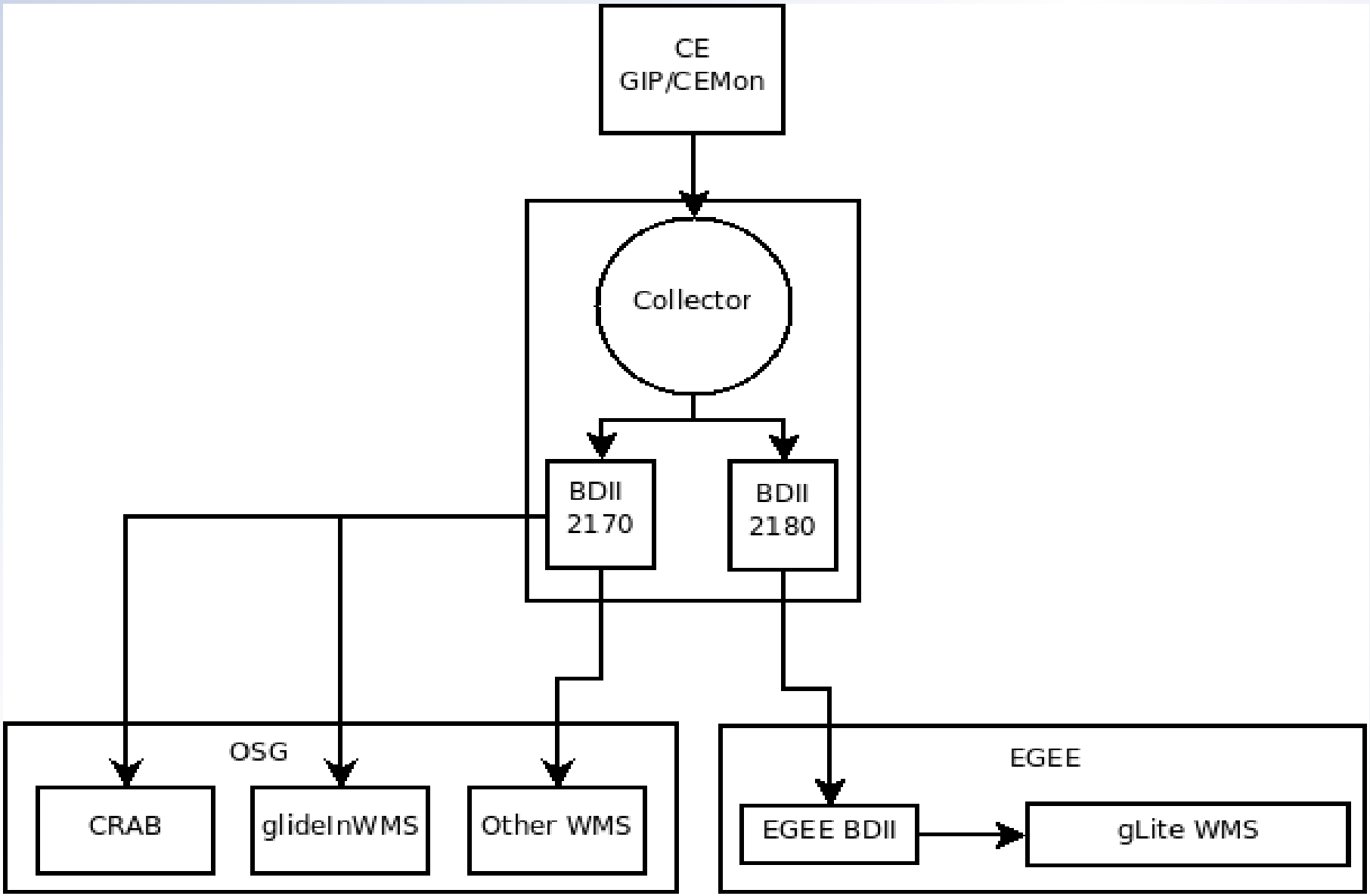
RESS Information Flow



RESS – Pros & Cons

- Pros
  - Very Simple for VO start up
  - Small VO's won't need a full WMS to run jobs
  - Limited hardware/software investment for VO's
    - Need Condor
    - Submit jobs via Condor-G
- Cons
  - Limited to Condor
  - Infrastructure is very fragile
    - A small (schema legal) change could cause an unmanageable amount of classads to be generated
    - Single point of failure – cannot distribute this service

BDII Information Flow



BDII – Pros & Cons

- Pros
  - Interoperability with EGEE
  - Somewhat scalable
  - Very visible information
    - Via ldapsearch
    - Via collector web interface
  - Well defined data structure (Glue Schema)
- Cons
  - Requires code to interpret queries – LDIF is not very human readable
  - Infrastructure is fragile
    - Custom attributes are difficult to add
    - BDII can be overloaded fairly easily

ATLAS

- ATLAS has a different way of accessing information about resources.
  - Everything runs through PANDA
    - PANDA has an internal database of known sites and queues
    - All jobs get matched through the internal database
  - PANDA's database gets updated in two fashions:
    - For EGEE sites ATLAS queries the EGEE BDII exclusively
    - For US sites, the process is more tedious
      - A list of hard-coded information is maintained
      - VORS is queried, but information from VORS does not override any hard-coded information
      - Any updates are handled manually
      - Some limited BDII queries are performed

Strawman Proposal Assumptions

- The current infrastructure is inadequate
- To meet the goals for OSG usage and usage expansion the End to End Information Infrastructure must be changed
- OSG wants to integrate with more grids (already interoperable with EGGE)
- The time frame for any changes is not defined

Strawman Proposal (1) - from Ted Hesselroth's email

- Integrate with OGSA-DAI
  - Proxy server performs queries in native format, uses workflows to convert to XML and caches as XML.
  - Search in OSG via XPath or XQuery.
  - Pros
    - Mature, well-documented in wide use in Europe.
    - Supports SQL, XPath, and file searches.
    - Modular, extensible. Could add ldap search as a module.
  - Cons
    - Had considerable funding, development may have slowed.
    - Not widely used in US.
    - To avoid wholesale replication of databases, would require on-the-fly conversion of XPath queries to native formats.
    - Adds an external software layer that increases complexity and possible bottlenecks

Strawman Proposal (2)

- Integrate with MDS4
  - Index Service queries WSRF services on their "resources".
  - Supports XPath only. All data are in XML.
  - Pros
    - Option to call arbitrary code to collect information
    - Mature Code
    - Wide use in US
    - Deep hooks within Globus software
    - Already have a relationship with the developing institution
  - Cons
    - Documentation needs work
    - May need to upgrade version of Globus in VDT to use full capability
    - Scalability questions?

Strawman Proposals

- For Either Proposal:
  - Both proposals include client and server software
  - Need to write “adapters” to collect and expose custom information
  - Need to be careful of scope, or support for the end product may be difficult
  - The internal “lingua franca” changes to XML
- MDS4 Vs. OGSA-DAI
  - MDS4 is deeply integrated with all Globus components already while OGSA-DAI exists as a separate set of services
  - MDS4 has US presence (LIGO, TeraGrid, etc.)

Administrative Concerns

- Both MDS4 and OGSA-DAI appear to use a “pull” model rather than the current “push”
  - Adds a layer of complexity to debugging activities
- May need to change the packaging of Tomcat to allow for “application management”. I don't believe this is enabled out of the box
  - The reason for this is that as we add more services to Tomcat, simply stopping and starting Tomcat may not be a viable solution to nagging problems (like CEMon).
- New expertise will be required at all levels for either technology. (Development, Architecture, Service Administration)

Information Consumption Options

- Make any architecture changes transparent to the VO
  - Effort will be required to write adapters
    - BDII
      - convert xml to ldif
      - Determine how we want to serve the ldif (use existing infrastructure?)
    - RESS
      - Convert xml to condor classads?
      - Pipe raw xml to the IG on RESS and have RESS perform translation?
- Make VO's change
  - Requires buy-in from the VO (not likely)
  - Example: CMS will still need to publish information to the EGEE BDII infrastructure